A New Shared Segment Protection Method for Survivable Networks with Guaranteed Recovery Time

János Tapolcai, Member, IEEE, Pin-Han Ho, Member, IEEE, Dominique Verchère, Member, IEEE, Tibor Cinkler, Member, IEEE, and Anwar Haque, Member, IEEE

Abstract—Shared Segment Protection (SSP), compared with Shared Path Protection (SPP), and Shared Link Protection (SLP), provides an optimal protection configuration due to the ability of maximizing spare capacity sharing, and reducing the restoration time in cases of a single link failure. This paper provides a thorough study on SSP under the GMPLS-based recovery framework, where an effective survivable routing algorithm for SSP is proposed. The tradeoff between the price (i.e., cost representing the amount of resources, and the blocking probability), and the restoration time is extensively studied by simulations on three networks with highly dynamic traffic. We demonstrate that the proposed survivable routing algorithm can be a powerful solution for meeting stringent delay upper bounds for achieving high restorability of transport services. This can significantly improve the network reliability, and enable more advanced, mission critical services in the networks. The comparison among the three protection types further verifies that the proposed scheme can yield significant advantages over Shared Path Protection, and Shared Link Protection.

Index Terms—Complete routing information scenario, integer linear program (ILP), segment shared protection (SSP), shared risk group (SRG), single failure scenario, working and protection paths.

ACRONYM1

SSP Shared Segment Protection
SPP Shared Path Protection
SLP Shared Link Protection
GMPLS Generalized Multi-Protocol Label Switching

ASON Automated Switching Optical Network
SDH Synchronous Digital Hierarchy
SONET Synchronous Optical Networking
LSP Label Switched Paths
PSC Packet Switching Capable
TDM Time-Division Multiplexing
OXC Optical Cross Connect
SRG Shared Risk Group
GSP General Shared Protection
ILP Integer Linear Program

NOTATION

\( W \) working path
\( W_k \) \( k \)-th working segment
\( P_k \) \( k \)-th protection segment
\( k_{\text{max}} \) maximum number of segments
\( G \) network topology with a set of arcs \( A \), and vertices \( V \)
\( c_j \) cost for allocating a capacity unit on arc \( j \)
\( f_j \) unreserved free capacity along arc \( j \)
\( v_j \) amount of spare capacity reserved for protection along arc \( j \)
\( S \) spare provision matrix
\( s_{\text{ea}} \) an entry in the matrix \( S \) assigned to a link pair
\( t_r \) recovery time
\( t_d \) link failure detection time
\( t_n \) notification time
\( t_a \) shared segment activation time
\( t_c \) data traffic restoration completion number of TDM LSP interrupted by the link failure
\( n \) number of upstream OXC nodes
\( d_i \) failure indication signal processing time
\( d_n \) time of forwarding the failure notify message
\( l_i \) length of the \( i \)-th link in kilometers
\( d_p \) propagation delay on the optical links
\( d_c \) time to complete the cross-connection
\( d_s, d_m \) time required to process the signaling message
\( t_0 \) permanent recovery time
\( \xi^w, \xi^p \) link overhead parameters

1The singular and plural of an acronym are always spelled the same.

Manuscript received December 27, 2006; revised July 7, 2007; accepted August 7, 2007. This work was supported in part by the EU FP6 IP NOBEL (http://www.ist-nobel.org) framework, the Bell University Labs (BUL) Program, High Speed Network Laboratory (HSNLab), the Hungarian National Research Fund, and the National Office for Research and Technology (Grant Number OTKA 67651). The work of J. Tapolcai was supported by the “János Bolyai Research Scholarship” of the Hungarian Academy of Sciences and by Öveges József Program of Hungarian National Office for Research and Technology. Associate Editor: H. Li.

J. Tapolcai and T. Cinkler are with the Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, Budapest 1117, Hungary (e-mail: tapolcai@tmit.bme.hu; cinkler@tmit.bme.hu).

P.-H. Ho is with the Department of Electrical and Computer Engineering, University of Waterloo, ON N2L 3G1, Canada (e-mail: pinhan@hph3.uwaterloo.ca).

D. Verchère is with Alcatel, 91460 Marcoussis, France (e-mail: Dominique.Verchere@alcatel.com).

A. Haque is with Bell Canada, Canada (e-mail: anwar.haque@bell.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRAN.2008.923480

© 2008 IEEE
Recently, the Internet Engineering Task Force (IETF) has made a lot of contributions to extend its Generalized Multi-Protocol Label Switching (GMPLS) protocol suite to satisfy the requirements of the Automated Switching Optical Network architecture (ASON) specified by the International Telecommunication Union Telecommunication Standardization Sector (ITU-T). On the one hand, the GMPLS framework specifies all the protocol capabilities in terms of signaling such as Resource reSerVation Protocol with Traffic-Engineering extensions (RSVP-TE), routing such as Open Shortest Path First with Traffic Engineering extensions (OSPF-TE), and link management such as Link Management Protocol (LMP). On the other hand, ITU-T ASON recommendations specify the architecture and requirements for the Automatic Switched Transport Network as applicable to carrier transport networks.

The label switching concept introduced in the Multi-Protocol Label Switching (MPLS) architecture is generalized to all switching technologies in the Generalized Multi-Protocol Label Switching (GMPLS) architecture. A label can be defined as a synthesis of connection-oriented, and connectionless technologies. GMPLS extends the label switching capabilities to network elements with a non-packet based forwarding engine, from packet switching capable (PSC) technologies to fiber switching capable (FSC) technologies including Layer 2 switching capable (e.g. Ethernet, ATM, Frame Relay), time division switching capable (e.g. SDH/SONET, PDH, and OTN), lambda switching capable (e.g. wavelength including CWDM and DWDM, and waveband), and fiber switching capable (e.g. ports). Within the GMPLS architecture, the connections between the ingress and egress nodes are referred to as Label Switched Paths (LSP). The traffic flowing along a LSP is defined by the label applied at the ingress node.

GMPLS has been recognized as the most promising control plane framework for the next generation carrier networks enabling a uniform, simpler management function for heterogeneous networks. One key requirement of network service providers for designing a transport network is to ensure survivability in the case of a network failure. Network outages are caused by a variety of events that can lead the network status into unpredictable states. The survivability refers to the ability to maintain the consistent service level agreement in the case of a network failure. The survivability can rely on the uses of different recovery mechanisms (i.e. protection, and restoration) implemented in the transport network. GMPLS-based recovery is a suite of failure protection and restoration mechanisms defined under the GMPLS framework, which is expected to provide complete solutions for achieving Quality of Service (QoS)-aware protection and restoration in a Data-centric heterogeneous network. The GMPLS-based recovery mechanisms are typically categorized in functions of the recovery moment. Recovery mechanisms are typically categorized based on the time, relative to the time of failure, at which path computation, signaling, resource selection, and allocation are performed. This grouping is illustrated schematically in Fig. 1. The term “Dynamic re-routing” is used when the working connection (or equivalently working LSP) is dynamically recoverable using (non pre-planned) ingress-egress re-routing. Pre-planned re-routing without extra traffic is used when protecting LSP are provisioned at the control plane level only (a.k.a. soft-provisioned). Finally, the term “Protection” is used if the protecting connections are either fully provisioned, thus allowing for extra-traffic transport, and precluding sharing of recovery resources; or if the protecting connections do not allow sharing of the recovery resources nor the transport of extra-traffic (due to the duplication of the signal over both working, and protecting connections).

During the recovery phase, soft-provisioned protecting LSP requires a signaling message sequence to allocate the transport network resources to the protecting LSP to be able to carry data traffic. Then this resource allocation sequence needs additional time by comparison to dedicated protecting LSP schemes such as 1+1 bi-directional protection, or 1:1 protection with Extra-Traffic. For a given GMPLS-based recovery mechanism, three main different extents of protection can be configured [1]: span

\[ x_{e}, y_{e} \] binary flow indicators of the working, and protection path of the ILP formulation
\[ A_{w}, A_{p} \] set of residual arcs in the network topology used to solve the working, and protection path segments
\[ c_{ab}^{\text{up}} \] an upper bound on the cost of the solution
\[ c_{pre}^{\text{up}} \] cost of the relaxed solution
\[ c_{SP} \] cost of the derived shortest path
\[ \xi_{i,j}^{\text{w}} \] detour factor of each arc assigned to the connection & network state
\[ s_{p}(G, i, j) \] cost of the shortest path in \( G \) between node \( i \), and \( j \)
\[ c_{BP} \] lower bound on the cost of protection paths
\[ \xi_{\text{max}} \] upper bound on the detour of the optimal solution
\[ m_{e} \] upper bound on the sharable spare capacity on link \( e \)
\[ \xi_{e}^{\text{p}} \] detour factor of each arc in \( A_{p} \)

Fig. 1. Classification of GMPLS recovery mechanisms.
(i.e. link), segment, or path-oriented protection schemes. Each protection scheme imposes a specific set of design principles, and requirements on the spare capacity allocation.

This paper focuses on segment LSP recovery mechanisms, classified as Segment Shared Protection (SSP), which has been proved to be able to achieve better capacity-efficiency; and more flexible resource allocation for meeting diversified design requirements, such as restoration time, and connection reliability. SSP has been widely studied [2]–[7] through heuristic approaches in locating the switching/merging node pairs, along with the working path segments, and protection path segments. The wide employment of heuristic approaches is due to the extremely high computation complexity, and large design space in locating an optimal routing solution. The study in [8] has provided an Integer Linear Program (ILP), which can find the optimal configuration, and spare capacity allocation for implementing the SSP. However, solving the ILP formulation is extremely time-consuming, which seriously limits its applicability. In this paper, a novel survivable routing approach for realizing SSP is developed based on the ILP formulation [8]. Instead of incurring a huge computation complexity in solving an ILP, or performing dynamic programming, our approach is to derive a good approximation on some of the important parameters in the ILP proposed in [8] by referring to the result of solving any heuristic SPP or SSP algorithm such that a significant reduction on the design space can be achieved. We will demonstrate in the simulation that the least-cost solution can be derived in most of the cases with our approach.

With the proposed algorithm, experiments are conducted to evaluate the performance impairment by tightening the restoration time constraints in both European, and North-American reference networks with dynamic connection arrival and departure requests. Two performance metrics are adopted: the success rate for establishing a connection with SSP, and the average cost in terms of network resources (i.e. bandwidth requirements). Based on the two performance metrics, a comparison among Shared Segment Protection (SSP), Shared Link Protection (SLP), and Shared Path Protection (SPP) under different restoration time constraints is made.

This paper is organized as follows. In Section II, we define the problem of SSP along with the associated modeling techniques. In Section III, we present the heuristic approach to efficiently compute the ILP formulation for proposing the optimal SSP solutions including the constraints on restoration time. We present the original SSP ILP [8] in Appendix A. In Section IV, we evaluate the performance of the proposed algorithm, and compare the results with that by SLP, and SPP. Section V concludes this paper.

II. BACKGROUND

A. Problem Definition

SSP is characterized by flexible configuration, and manipulative recovery time for different failure events, as well as efficient spare capacity consumption [2]–[8], as compared with all the other types of protection. The segment protection configuration can recover connection disruption over a portion of a connection, where the restoration of a connection can be performed locally. As a result, the recovery time of a failure can be guaranteed by manipulating the length of the working, and protection path segments.

Dynamic survivable routing for SSP is investigated in this paper, where the connection requests arrive at the network one after the other without any knowledge of future arrival. The traffic pattern, nonetheless, is defined through the arrival & departure rate for each source-destination pair, which is in turn determined by the historical record of traffic distribution provided in [9], [10]. The objective of the proposed algorithm is to find a working path (denoted as $W$) corresponding to a connection request, the switching/merging node-pairs which define segments along $W$, and a protection path segment corresponding to each segment along $W$ (see also Fig. 2). Here, the $k$-th working segment (denoted as $W_k$), and the $k$-th protection segment (denoted as $P_k$) must be link disjoint for $\forall k$.

Take a graph $G(V, A)$ representing the network with a set of arcs $A$, and vertices $V$, where $|A|$, and $|V|$ are the number of arcs, and vertices respectively in $G$. In the non PSC context (e.g. TDM, lambda or Fiber), a “transport link” is bi-directional interconnecting two nodes. In the graph representing the network, a link is modeled as two oppositely directed arcs between the same edge nodes. The cost for allocating a capacity unit on arc $j$ is denoted as $c_j \forall j \in A$. The unused free capacity along arc $j$ is denoted as $f_j \forall j \in A$. The amount of spare capacity reserved for protection along arc $j$ is denoted as $v_j \forall j \in A$. Let the source, and the destination of the upcoming connection request be respectively denoted as $s$, and $d$, both with a bandwidth of $b$. The failures assumed in this paper are limited to a link cut, and no more than a single link in the graph can be possibly hit by failure at any moment. However, the most general definition of Shared Risk Group (SRG) [11] can be easily applied to the proposed algorithm.

In [12], we introduce a spare provision matrix, which is denoted as $S$ with a size $|A| \times |A|$. Each entry in the matrix is denoted as $s_{e,a}$, where $e = 1 \ldots |A|$, $a = 1 \ldots |A|$. Each entry $s_{e,a}$ represents the amount of non-sharable spare capacity along arc $e$ for the protection path segment if the corresponding working segment is involved in the $a$-th arc (see also Fig. 3). The non-sharable spare capacity should be allocated along protection routes so as to sufficiently recover the working bandwidth affected by any failure event. The condition for routing $W$ with a bandwidth $b$ along arc $a$ is $f_a \geq b$, while the feasible condition for the $k$-th protection path segment to pass through arc $e$ is $f_e + v_e \geq b + \max_{a \in W_k} s_{e,a}$ for $\forall k$ [8].

![Fig. 2. An illustration of SSP. The working path is $s = b - c - d - e - f - g - h - i - j - k - l$, which is divided into three segments: $a = b - c - d$, $d - e - f - g - h$, and $h - i - j - k - l$.](image-url)
B. System Formulation

We describe the GMPLS failure restoration mechanisms in this section, along with the corresponding recovery time model adopted in the study.

1) Recovery Time Model: Based on the MPLS recovery time sequence diagram of [13], a GMPLS segment recovery time model is defined as specified in [14]. The recovery time \( (t_r) \) is composed of four terms: the link failure detection time \( (t_d) \), the notification time \( (t_n) \), the activation of the shared segment and the cross-connection time \( (t_c) \), and the data traffic restoration completion time \( (t_d) \). The failure detection time \( (t_d) \) depends on the data plane failure detection mechanism, and the failure report time to the GMPLS controller; \( t_d \) is taken as 10 ms in the context of an SONET/SDH transport network.

The failure notification time \( (t_n) \) is the total time during which the failure notification signaling message is built at the node adjacent to the link failure, and then sent to the node responsible to recover the failed TDM LSP, i.e. the branch-node. Each failure notification signaling message is forwarded towards a branch node. Each failure notification message carries the signaling session information of all the failed TDM LSP to be recovered by this branch node. In this model, it is supposed that the failure notification messages follow the same path as the failed TDM LSP.

The TDM LSP recovery time \( (t_r) \) is the time required to signal the allocation of the pre-reserved resources of the protecting segment at the transport level. Each time a signaling message reaches the GMPLS node controller, the controller orders the switching fabric to process the cross-connection. Before the cross-connection is completed, the node sends out the signaling message to the next node. The ultimate node is the merge node. Finally, the traffic of the failed working LSP is recovered after a propagation time \( (t_c) \) from the branch node to the merge node. The total recovery time \( (t_r) \) can be modeled as

\[
t_r = t_d + t_n + t_a + t_c \quad \text{such that}
\]

\[
t_r = t_d + t_n + \left[ f \cdot d_l + n \cdot d_n + \sum_{i=1}^{n-1} l_i \cdot d_p \right]
+ \left[ P_k \cdot d_s + \sum_{i=1}^{P_k} l_i \cdot d_p + d_m \cdot P_k \cdot d_c \right] + \left[ P_k \cdot d_s \right]
\]

where \( f \) is the number of TDM LSP interrupted by the link failure, and for simplicity reasons it is limited to a maximum of 100 LSP in this study. Variable \( n \) denotes the number of optical cross connect (OXC) nodes from the upstream node adjacent (including it) to the failure to the branch node of the recovery segment. The link failure detection is performed at the Line Card interface of the SDH/GMPLS Cross-Connect. Once the link failure is declared at the Line Card by the system fault control functions (data plane), the link failure event is reported to the GMPLS controller. The GMPLS controller finite state machine (FSM) enters the Faulty State [15]. The link failure detection time and report at the GMPLS node controller is supposed to be fixed (i.e. constant) in this model. The failure indication signal processing time at the adjacent node interface is denoted by \( d_i \). It is set to \( 10 \mu s \) per TDM LSP. Similar to [13] for the \( f \) TDM LSP, the total time required to initiate the failure notify message is \( f \cdot d_i \).

The failure notify message is not processed by each GMPLS controller, but it is forwarded at each intermediate node it goes through. At each node, the time required to forward the failure notify message is denoted by \( d_n \). In the segment recovery scheme, the failure notify message goes from the upstream node that is adjacent to the link failure towards the branch node. In this model, the time of forwarding the failure notify message \( d_n \) is set to 1 ms per node. The length of the \( i \)-th link of the segment after the branch node, and before the merge node, is denoted by \( l_i \); and the propagation delay on the optical links is denoted by \( d_p \), set to \( 5 \mu s \) per km.

We assume that the link failure indication messages of the affected TDM LSP are forwarded sequentially at each node interface. Therefore, the evaluation of the upper bound on the link failure notification time must consider the summation of the three terms \( [f \cdot d_i + n \cdot d_n + \sum_{i=1}^{n-1} l_i \cdot P_k \cdot d_p] \). The term \( [P_k \cdot d_s] \) is the number of links on the \( k \)-th protecting segment of the failed \( k \)-th working segment; take as an example the number of links on the \( k \)-th protecting segment between the recovery branch node, and the recovery merge node. The time required to process the signaling message that triggers the cross-connection of the switching fabric of the OXC at each intermediate node from the branch node to the merge node is denoted by \( d_m \). This time is set to 10 ms per node.

Finally \( d_n \cdot [P_k] + d_c \) is the total time to configure, test, and setup the switching matrix of each OXC node (switching time) along the protecting segment route. \( d_c \) is the time to complete the cross-connection of the merge node. \( d_m \) is assumed to be 10 ms, and \( d_c \) is 20 ms.

Eq. (1) can be re-ordered as

\[
t_r = (t_d + t_n + f \cdot d_i + d_c) + \left[ n \cdot d_n + \sum_{i=1}^{n-1} l_i \cdot P_k \cdot d_p \right]
\]

\[
+ \left[ P_k \cdot d_s + \sum_{i=1}^{P_k} l_i \cdot d_p + d_m \cdot P_k \cdot d_c \right] + \left[ P_k \cdot d_s \right]
\]

\[
(2)
\]

The expression \( t_d + t_n + f \cdot d_i + d_c \) is assumed to be less than \( t_0 \), where \( t_0 = 21 \text{ ms} \). Two link overhead parameters are defined as \( c_i = d_i + l_i \cdot d_p \), and \( c_i = d_a + 2 \cdot l_i \cdot d_p + d_m \) for each link \( i \); and because \( n \leq [W_k] \), (2) can be further rewritten as

\[
t_r < t_0 + \sum_{i=1}^{P_k} c_i + \sum_{i=1}^{P_k} c_i
\]

A global parameter \( \bar{c}_{max} \) is the maximum allowable recovery time for the corresponding LSP defined in the service level
agreement, and is the maximum restoration time of all the self-healing units.

Observe that the recovery time of a link failure event strongly depends on the length, and hop counts of the working, and protection path segments, which, on the other hand, serve as an important factor to determine the survivability through a dual failure event for the connection. Note that, with SSP, a connection cannot be restored through a dual failure event only when the two failures hit both working, and its protection path segments. This can be well managed under our framework because, with the recovery time constrained, the length (or the hop count) of the working and protection segments will be constrained accordingly, which will subsequently ensure the dual failure survivability for the connection. Thus, the overall service availability of the connection will also be constrained.

III. THE SSP ALGORITHM

A. An Extended ILP Formulation

An overview on the ILP formulation for implementing ILP for an optimal (or least-cost) SSP solution is given in the Appendix, and in [8]. The solution yields the least-cost working and protection path segments, and the allocation of the switching & merging node-pair of each working & protection path segment pair. To address the constraint on the restoration time for a connection, the total length of each working & protection path segment pair for the connection must be upper-bounded. Therefore, the following constraint is appended:

$$\sum_{\forall a \in A_w} c^w_a \cdot x^w_a + \sum_{\forall e \in A_p} c^p_e \cdot y^p_e \leq \zeta_{\text{max}}$$

for $1 \leq k \leq k_{\text{max}}$

(4)

The Appendix provides detailed definitions on these residual network graphs. Eq. (4) brings $k_{\text{max}}$ additional constraints to the ILP formulation. Note that the optimal SLP can be derive by adding (5) as additional constraints to the ILP formulation of SSP.

$$\sum_{\forall a \in A_w} x^w_a \leq 1 \text{ for } 1 \leq k \leq k_{\text{max}}$$

(5)

B. The Proposed Algorithm

The proposed algorithm is characterized by the intelligent removal of arcs in the network topology which are unlikely to be adopted by the protection path segments. This is achieved by a pre-calculation mechanism, which defines the detour factor for each arc. The arcs with a larger detour factor are not likely to be taken by the protection path segments because they are generally far from the source and destination nodes. With the reduced network topology, the solving ILP can be significantly faster at the expense of an increase in the total cost of the connection. In other words, by manipulating the threshold of detour factor in the arc removal process, a compromise between the quality of the solution and the calculation time can be adjusted.

In addition, because we have observed that the computation complexity strongly depends on the maximum number of segments (i.e., $k_{\text{max}}$) to be considered for each connection, the proposed algorithm is equipped with a more adaptive approach in selecting the value of $k_{\text{max}}$, which is expected to significantly improve the computation efficiency. The above two devices make the proposed algorithm contributive to the state-of-the-art in solving the SSP survivable routing problem.

Fig. 4 shows an illustration of the main phases of the proposed algorithm. First, a pre-calculation mechanism is devised to calculate a feasible (or a good approximation on the) solution, which gives an upper-bound on the total cost. Second, based on the pre-calculated solution, it estimates the value of $k_{\text{max}}$, and excludes those arcs relatively unlikely to be taken by the working, and protection segments. The graphs for solving W, and Q are denoted as $A_w$, and $A_p$ respectively. Finally, the ILP SSP is launched on the reduced residual graphs with the proper $k_{\text{max}}$ Parameter.

Fig. 5 shows the flowchart on the first phase of the proposed algorithm, where an upper bound on the cost of the solution (denoted as $\zeta_{\text{ub}}$) is derived by any arbitrary heuristic algorithm for SSP. In this paper, we derive an arbitrary SSP solution using the following approach. The ILP problem for SPP [16] is solved with an additional constraint on restoration time:

$$\sum_{\forall (a) \in A} c^w_a \cdot x_a + \sum_{\forall (e) \in A} c^p_e \cdot y_e \leq \zeta_{\text{max}}$$

(6)

where $x_a$, and $y_e$ are binary flow indicators of the working, and protection path, respectively. If the working path intersects with the protection path, the nodes of intersection are simply
treated as branch & merge nodes, by which a feasible solution is identified.

At step 1, the algorithm tries to find a feasible solution, and a valid upper bound on the total cost. If the length of the working path is one hop, then the searching is terminated in step 3. Otherwise, the cost of the feasible solution is stored as an upper bound in \( c^{ub} \), and the number of segments of the feasible solution is set as \( k_{max} + 1 \).

If no feasible solution of SPP is derived in step 1, a heuristic approach is developed to define \( c^{ub} \) described as follows. In step 4 of the flowchart, we solve the SPP again with the restoration time constraint relaxed (i.e., (6) is eliminated from the ILP of [16]). If we find a shortest path in graph \( G(V, A_W) \), we go to step 5, and define \( c^{ub} = 2 \cdot c^{pre} \), where \( c^{pre} \) is the cost of the relaxed solution. In this case \( k_{max} \) is set as the number of hops in the derived working path. If solving the SPP with the relaxed restoration time constraint still fails, we go to step 6, and calculate the shortest path; and in (7), we define \( c^{ub} = 3 \cdot c_{sp} \), where \( c_{sp} \) is the cost of the derived shortest path. We estimate \( k_{max} \) similarly as the number of hops in the shortest path. Obviously, if the shortest-path algorithm fails to find any solution, we go to step 8 to terminate the algorithm. Fig. 6 shows the second phase of the algorithm, where the graphs \( A_{W} \), and \( A_{P} \), are derived as follows. Let us define a detour factor of each arc assigned to the connection & network state, denoted as \( \xi_{e}^{w} \), which can be calculated by the formula

\[
\xi_{e}^{w} = sp(G_{w}, s, i) + c_{i,j} + sp(G_{w}, j, d)
\]

where \( sp(G_{w}, s, i) \) represents the cost of the shortest path in \( G_{w} \) between \( s \), and \( i \); \( c_{i,j} \) represents the cost of arc \( (i, j) \); and \( sp(G_{w}, j, d) \) is the cost of the shortest path between \( j \), and \( d \) (see also Fig. 7).

The detour factor shows the minimal detour compared with that of the shortest path if the working path passes through arc \((i, j)\). Let us define \( c_{lb}^{p} \) as the lower bound on the cost of the protection paths. \( c_{lb}^{p} \) is set to 0 at the beginning of the phase. Let us define \( \xi_{max}^{w} \), which gives an upper bound on the detour of the optimal solution. In the first step, \( \xi_{max}^{w} \) can be set to \( c_{lb}^{p} = sp(G_{w}, s, d) - c_{lb}^{p} \) in the case that the pre-calculated solution was feasible. As a result, all the arcs with \( \xi_{e}^{w} > \xi_{max}^{w} \) can be removed from \( A_{w} \). Another necessary condition on each arc is that the failure of arc \( a \) can be restored by the capacity (including the amount of free & sharable spare capacity) along arc \( e \). Each arc can be restored if after the failure there is a feasible protection path.

**Definition:** The General Shared Protection (GSP) test of arc \( a \) is true if there is a path \( P \) between \( s \) and \( d \) which is link-disjointed from arc \( a \) such that \( \forall e \in P, f_{e} + v_{e} - s_{e, a} \leq b \).

It is easy to verify that the working route has only arcs where the GSP test holds. As a result, in step 11 of the flowchart, \( A_{w} \) contains all the arcs with \( \xi_{e}^{w} \leq \xi_{max}^{w} \) and \( b \leq f_{e} \), and those which satisfy the GSP test.

After \( |A_{w}| \) is reduced, the lower bound on the cost of the protection path can be derived. First, a lower bound on the cost of each arc taken by protection path is calculated. Obviously, the working path will take at least one arc of \( A_{w} \). Because \( A_{w} \subseteq A \), a better upper bound \( m_{e} \) on the sharable spare capacity on link \( e \) can be derived as shown on step 12 of the flowchart:

\[
m_{e} = \min_{a \in A_{w}} v_{e} - s_{e, a}
\]

With the information of \( m_{e}, A_{p} \) can be defined as the graph containing all the arcs with \( b \leq v_{e} + m_{e} \). Thus, a lower bound on Fig. 5. The flowchart of pre-calculation of an upper bound on the total cost, and estimating parameter \( k_{max} \).

Fig. 6. The flowchart of the second phase of the pre-calculation process, which reduces \( A_{w} \) and \( A_{p} \), while the optimality of the result is still guaranteed.

Fig. 7. An illustration of the detour factor \( \xi_{e}^{w} \).
the cost of each arc taken by the protection path can be determined as

\[ c_{i,j}^p = \max\{b - v_e + m_e, 0\} \cdot c_e \]  

(9)

The above relation holds because \( v_e - m_e \) is a lower bound of the sharable spare capacity such that the term \( \max\{b - v_e + m_e + b, 0\} / b \) gives a lower bound on the ratio of the sharable capacity. With this, we can derive the detour factor of each arc in \( A_p \) (denoted as \( \xi_i^p \)), which is specific to the connection request, and the current link-state.

\[ \xi_i^{p,j} = sp(G_p, s, i) + c_{i,j}^p + sp(G_p, j, d) \]  

(10)

where \( sp(G_p, s, i) \) and \( sp(G_p, j, d) \) represent the total cost of the shortest path in graph \( G_p \) between the source node and node \( i \), and in graph \( G_p \) between node \( j \) and the destination node, respectively; and \( c_{i,j}^p \) is the cost of \((i, j)\) in \( G_p \). With \( sp(G_p, j, d) \), we can derive a better lower bound on \( c_{i,j}^p \). Note that the arcs with \( \xi_i^p > \xi_i^{\max} \) are not included in \( A_p \) because the desired protection path can not feasibly pass through any of them due to its large amount of detour. Therefore, as shown in step 14 of the flowchart, \( A_p \) is composed of all the arcs where \( \xi_i^p \leq \xi_i^{\max} \), and \( b \leq f_e + m_e \). If the lower bound on the cost of the protection path \((c_{i,j}^p)\) has been improved, the algorithm switches back to step 10 of the flowchart to further reduce the arcs of \( A_W \), and \( A_p \) in the next iteration.

With the proposed algorithm, the runtime of solving the ILP of SSP can be significantly reduced, while the quality of the result is almost always guaranteed or very close to the optimal one. We will further verify the heuristic in the following section.

IV. VERIFICATION

Experiments are conducted to verify the proposed algorithm using CPLEX 8.0 on a Sun Ultra 80 workstation with 2GB memory, and several Linux workstations. As a comparison among SLP, SPP, and SSP, simulations are conducted on two network topologies. The first one is the pan-European fiber-optic network defined by IST project LION & COST action 266 as [17], which has 28 nodes, and 57 bi-directional links as shown on Fig. 8. The second one is based on the US NSF Network [18] with 26 nodes, and 43 bi-directional links as shown in Fig. 9. For both networks, a traffic matrix for year 2005 is estimated according to [9], which is a slightly improved model than that provided in [10]. A dynamic traffic pattern is generated according to the traffic matrix for year 2005 such that an Interrupted Poisson Process, and Pareto inter-arrival times are integrated together with exponential holding time. Simulation is arranged as follows to investigate the impact of the restoration constraint. For simplicity in the restoration time constraint, \( \zeta_i^W = 1 \), and \( \zeta_i^p = 1 \) were assigned for all arcs \( a \). Therefore, the restoration constraint parameter \( \zeta_i^{\max} \) represents the hop count of the working, and protection segment-pair of each protection domain. For each connection request, the ILP is solved with different values of \( \zeta_i^{\max} \) (i.e., the restoration time constraint). Fig. 10 shows the flowchart of the simulation process. For each connection request, a series of SSP survivable routing problems are solved with different values of \( \zeta_i^{\max} \), where two performance metrics are evaluated: one is the number of successful set-ups, and the other is the average cost using the target function of the ILP.

Fig. 11 shows two illustrative examples, where simulation is conducted on the network N16 (with 16 nodes, and 27 bi-directional links) with a heavy load (at average total network utilization of 73%). The graph with the selected \( s - d \) pair is also
drawn on the charts. The y-axis represents the cost of the connection yielded by the target function of the ILP. It is clear that, as the restoration time constraint is getting more relaxed, the performance impairment (in terms of the cost) for each connection request is reduced.

As a comparison, the optimal SPP is evaluated using the ILP formulation of Section III, where the derived result for each connection is marked by triangles on the charts. Note that the cost of solving the SPP is not smaller than that of the SSP case because SPP is a special case of SSP (with \( k_{\text{max}} = 1 \)). Shared Link Protection (SLP) is also a special case of SSP where each working segment consists of only one link.

A detailed overview on SLP, and SPP can be seen in [7]. The simulation results of SLP are marked with boxes on the charts, in which the length limitation on the protection path can be addressed using (6). The selected node-pairs are selected far from each other such that it illustrates the increase of the cost of SSP if we gradually sharpen the restoration time constraint. Fig. 13 shows the results using a 61-node network (shown in Fig. 12) with a light traffic load for comparing the three types of protection in terms of average cost, and the success rate under different restoration time constraints. Results of 100 random connection requests are averaged for each data. It is shown that an average of approximately 10% reduction in the cost can be achieved with SSP over the case of SPP if the restoration time constraint is relaxed to 13 hops. The average cost in the SSP and SLP cases increase when the restoration time constraint is sharpened, as it was expected. However, the average cost drops dramatically when the restoration time constraint is loose because at this moment most of the long connections (with large cost) are blocked, and only the short connections (with small cost) can be allocated.

It can also be observed that SSP can yield a much higher success rate for those connection requests under a tight restoration time constraint than that with SPP at the expense of taking moderately extra cost, as shown in Fig. 13. On the other hand, SLP yields the highest average cost in all cases with a close success rate with SSP, and is seen as less competitive with the other two types of protection. Thus, we conclude that SSP can achieve the best advantages over both of the other two types of protection.

We further extend the simulation study on ERNet & NARNet with the same traffic pattern, and simulation environment. The simulation results of NARNet with a high traffic load are shown in Fig. 14, where the major advantage demonstrated in using SSP is that the success rate outperforms that of the SPP case by 2 times or more at the expense of a little bit higher cost (as shown in Fig. 14b)). For SLP, the overall performance is far outperformed by the other two cases, although it can guarantee the shortest restoration time.

Fig. 15 demonstrates the runtime by the proposed SSP algorithm in solving SSP, SLP, and SPP survivable routing in the two network topologies. It can be observed that the proposed scheme can significantly improve the solving of the ILP in [8] with the pre-calculation. Note that even with the reduced network topologies, and pre-calculation mechanism, the problem is still NP-hard. However, the runtime can be reduced by more...
Fig. 14. Performance impairment by addressing the restoration time constraint with NARNet at high load.

Fig. 15. Runtime in solving the SPP, SLP with pre-calculation, SLP, SSP with pre-calculation, and SSP on N16, and ERNet.

Fig. 16. Average cost, and success rate of optimal SSP solution, and SSP with pre-calculation step. For SLP there is an average of 3.3 gap.

than 100 times, and we can further reduce the runtime by removing more arcs in the networks that are unlikely to be traversed by the protection paths.

On Fig. 16, the results of the ILP SSP, and the results of the proposed algorithm, are close to each other in the whole range of the restoration time limit. It demonstrates that the performance of SSP is not significantly degraded by reducing the design space at the pre-calculation stage. To summarize the simulation results, SSP gave a good compromise in terms of cost versus restoration time, while SLP requires an average of 10–20% additional capacity allocation compared to SSP or SPP.

V. CONCLUSIONS

This paper studies the optimal configuration for establishing the SSP on the working LSP, in which an ILP is formulated such that the branch-merge node-pairs in each Working & Protecting segment, and the corresponding least-cost Working & Protecting segment-pair for a LSP request, can be jointly determined in a single step. To improve the high computation complexity induced by solving the ILP, we propose a novel heuristic called the SSP algorithm, aiming to initiate a graceful compromise between the optimality, and the computation time by employing a novel iterative arc reduction pre-calculation mechanism. Different from the ILP in [8], the demonstrated formulation considers the restoration time constraints, which meets the practical requirement of the successful optical transport network restoration. We conducted extensive simulation efforts on NARNet, and ERNet to compare the Shared Link Protection (SLP), Shared Path Protection (SPP), and Shared Segment Protection (SSP) in terms of average cost, and the success rate of setting up connections. We observe that SSP can initiate a graceful compromise between average cost, and network availability under a wide range of restoration time constraints.

APPENDIX

A. The ILP Formulation for SSP

This section overviews the linear formulation for the segment shared protection problem presented in [8]. The main idea is to define a path $Q$, called the mass protection path, which defines the route of each protection segment, as well as the branch & merge nodes of the working path ($W$). Similar to Suurballe’s [19] algorithm, $Q$ is composed of the reversed links along the working path, and all the backup segments. A simple example is shown in Fig. 17, where $Q$ is (s-a-b-c-e-d). The first protection domain is formed by the working, and protection segments (s-c-b), and (s-a-b), respectively; while the second is formed by (c-b-d), and (c-e-d), respectively. We allow the overlapping between the working segments of two neighbor protection domains in order to explore the largest design space so as to guarantee the optimality of the derived solution. Note that $Q$ may contain loops to reflect the fact that spare capacity sharing can
happen between two protection segments of different protection domains. Variable $k_{\text{max}}$ is defined as a parameter of the ILP, and represents the maximum number of protection domains that can be possibly handled in the problem (it is $\leq n - 1$).

Three residual graphs are defined for solving this problem:

- $G_w(V, A_w)$ is composed of links with $b \leq f_{ai}$ for $a \in A_w$ (in the following formulas $x$ is a binary, $\delta$ is a real, and $\delta^k$ are $k_{\text{max}}$ binary variables assigned to arcs of $G_{ai}$) that describe the working segments.

- $G_p(V, A_p)$ is composed of all the links $b \leq f_{ai} + v_e$ for $e \in A_p$ (in the following formulas $y^k$ is $k_{\text{max}}$ binary, and $r'$ is a real variable assigned to the arcs of $G_{ai}$), and it describes the protection segments. We need this graph to record the spare link-state because working and protection paths take separate suites of link-state with shared protection.

- $G_p'(V, A_p')$ is composed of all the links in $A_p$ along with the links of $A_w$ in a reversed direction. The “reversed” arcs corresponding to $A_w$ in $A_p'$ are denoted as $(\delta^r_j)$, and all the others (corresponding to $A_p$) are $(\delta^r_j)$. The graph is assigned to $Q$, and handles the reverse arcs caused by the overlapping between $Q$ and $W$. In the following formulas, $y'$ is an integer, $\delta$ is a real number, and $y$ is a binary variable assigned to the edges of $G_{ai}$).

$x_{ai}$ and $y_{ak}$ are flow indicators of path $W$, and $Q$, and $y_{ai}$ is for the protection route of segment $k$. The objective function is

$$\text{Minimize: } \sum_{a \in A_w} b \cdot c_a \cdot x_{ai} + \sum_{e \in A_p} (c_e \cdot z_e + e \cdot y_{ae})$$

where $c_a$ is the cost per unit of working bandwidth to reserve arc $a$; and $z_e$ is for the protection segments, and represents the amount of capacity, which cannot be shared, yet need to be allocated for the protection paths.

**The constraints are as follows.**

We need the flow conservation constraint for the working (on variable $x$), and mass protection paths (on variable $y$). We need to formulate the above-mentioned properties of working, and mass protection paths. $x_{ai}$, and $y_{ai}$ will be exclusive in terms of the SRG they take. An arc can be taken by $y_{ai}$ in a reversed direction only if $x_{ai}$ passes through it. Besides, each reversed arc can be used only once because the algorithm only allows two working segments overlapped. Thus, $Q$ is SRG disjoint from $W$ except for the reversed arcs of $W$. Note that reversed arcs indicate the branch/merge nodes for each protection domain along $W$. A pair of variables, $\delta_{ai}$, and $y_{ai}$, is assigned to each link along $W$, and $Q$, respectively, such that the first link from the source has a label of 1; and if a protection domain ends or starts at a node, the labels of the following arcs will be increased by 1. This labeling method is similar to that proposed in [2]. This is done by modified flow conservation constraints, where the following four situations are considered for all vertices (except for the source and destination) taken by $W$: (a) $Q$ merges back to $W$; (b) $Q$ branches out of $W$; (c) $Q$ merges back and branches out of $W$; (d) otherwise. The amount of flow of $\delta_{ai}$ and $y_{ai}$ at vertex $i$ along $W$ increases by 1 for case of situations (a) and (b), increases by 2 for case of situation (c), and is unchanged otherwise. With link labels $\delta_{ai}$, and $y_{ai}$, path $W$ is divided into segments such that each link along it is covered by at least one protection segment. This effort introduces $k_{\text{max}} \cdot \lvert A_w \rvert$, and $k_{\text{max}} \cdot \lvert A_p \rvert$ arc-domain incidence binary variables denoted as $x_{ai}$ and $y_{ai}$, which is 1 if arc $a$ (or $e$) is traversed by the working, and protection segment of the $k$th protection domain, respectively. By observing Fig. 18, one can easily verify that the value of $y_{ai}$ on $Q$ of the first protection domain is 1; and in the second protection domain $y_{ai}$ is 3; and in the $k$th protection domain $y_{ai}$ is $2k - 1$; thus $y_{ai} = 1$ only when $y_{ai} = 2k - 1$. The value of $\delta_{ai}$ of arc $a$ taken by $W$ in the first protection domain is either 1 or 2, depending on whether or not there is one or more overlapped arcs between the working segments of the first, and the second protection domain; while on the links of the $k$th protection domain, we have $\delta_{ai} = 2k - 2$ on the non-overlapped links, and $\delta_{ai} = 2k - 1$ on the overlapped links of the $(k - 1)$th, and the $k$th protection domain; we have $\delta_{ai} = 2k - 1$ on
the overlapped links of \( k \)th, and \( (k-1) \)th protection domain. Please refer to Fig. 19 for an example demonstrating the definition of \( x_{k}^{E} \), and \( y_{k}^{E} \).

The last constraints that need to be defined are the SRG constraints setting the value of \( z_{c} \) defined in the target function. It is considered using \( \mathbf{S} \) matrix, such that when link \( a_{i} \) and \( c \) is taken by the working, and protection segments in the \( k \)th protection domain, respectively, the resultant amount of scaling (i.e., \( z_{c} \)) is at least \( b - v_{c} + s_{c,a_{i}} \).

References


János Tapolcai (M’05) received his M.Sc. (’00 in Technical Informatics), and Ph.D. (’05 in Computer Science) degrees in Technical Informatics from Budapest University of Technology and Economics (BME), Budapest, Hungary. Currently he is an Associate Professor at the High-Speed Networks Laboratory at the Department of Telecommunications and Media Informatics at BME. His research interests include applied mathematics, combinatorial optimization, linear programming, linear algebra, routing in circuit switched survivable networks, availability analysis, grid networks, and distributed computing. He has been involved in a few related European and Canadian projects (IP NOBEL; NoE e-Photon/One; BUL). He is author of over 30 scientific publications, and is the recipient of the Best Paper Award in ICC’06.

Pin-Han Ho (M’04) received his Ph.D. from ECE Department at Queen’s University in 2002. He joined the Electrical and Computer Engineering department in the U of Waterloo, Canada, as an Assistant Professor in the same year. His current research interests cover a wide range of topics in wired and wireless communication networks. He is the recipient of the Early Researcher Award in 2005; and the Best Paper Awards in SPECTS’02, ICC’05 Optical Networking Symposium, and ICC’07 Security and Wireless Communications Symposium.

Dominique Verchère (M’04) received a Ph.D. in Computer Science from Paris-Sorbonne University. Since 1998, he has been with Alcatel-Lucent Bell Labs, working first on IP-PBX control software, on data path processing, terabit-switching capacity systems (scheduler & buffer management). He works on resilience for optical systems, wavelength-division-multiplexing (WDM) networks with enhanced functionalities based on ASON/GMPLS specifications. He contributed in several European projects (EuroNGI, Tbbones, VOLIA and NOBEL). He is working/leading Architecture and Protocols for Grid Computing services offered by Optical Network with Ultra-high transmission capacity (CARRIOCAS project). He has published more than forty papers, and holds more than twenty-five patents. He is a Distinguished Member of Alcatel-Lucent Technical Academy.

Tibor Cinkler (M’99) has received the M.Sc. (’94), and Ph.D. (’99) degree from the Budapest University of Technology and Economics (BME), Hungary, where he is currently Associate Professor at the Department of Telecommunications and Media Informatics. His research interests focus on optimization of routing, traffic engineering, design, configuration, and resilience of IP, Ethernet, MPLS, OTN, and particularly of heterogeneous GMPLS-controlled WDM-based multilayer networks. He has been involved in numerous related European and Hungarian projects including NOBEL, and e-Photon/One; and he is member of ICC, GLOBECOM, ONDM, DRCN and other Program Committees. He is an author of over 170 refereed scientific publications, and of 3 patents.

Anwar Haque received the B.S., M.Sc., and M.Math degrees in Computer Science from the North South University, Bangladesh, the University of Windsor, Canada, and the University of Waterloo, Canada in 1997, 2001, and 2007 respectively. He has published more than 25 refereed technical papers. He has served as a reviewer, and TPC member for many international journals and conferences. His research interests include IP network QoS (Quality of Service), network design and planning with focus on survivability, and network security. He is currently employed at Bell Canada as Manager-Network Planning. He is a member of the IEEE.